Docket No.: 003829.P004
Express Mail No.: EM560888030US

UNITED STATES PATENT APPLICATION

FOR

A METHOD OF SECURE COMMUNICATION OVER A DISTRIBUTED NETWORK WITHOUT USING SECURE SOCKET LAYER

Inventor:

Kim F. Storm

Prepared By:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025-1026
(310) 207-3800

A METHOD OF SECURE COMMUNICATION OVER A DISTRIBUTED NETWORK WITHOUT USING SECURE SOCKET LAYER

BACKGROUND

Field of the Invention

[0001] The invention relates to network communication. More specifically, the invention relates to secure communication between network devices without the use of secure socket layer.

Background

[0002] With the proliferation of the internet, the need and desirability of managing devices over a distributed network in a secure manner continues to increase. Typical web browsers support Java, Java Script, frames and forms. However, the contents of such frames, forms, and even the Java Script passed between a web browser and a web server is typically freely visible to potential third parties snooping the web traffic. To ensure proper management and to avoid intentional and unintentional acquisition of sensitive data by third parties, the exchange between a browser and a device under management, should be secure, e.g., both authenticated and encrypted.

[0003] To permit secure communication between network nodes, Secure Socket Layer ("SSL") was developed by Netscape Communications Corporation as a protocol to permit encrypted communications. SSL is layered under Hypertext Transfer Protocol ("HTTP") and Above Transmission Control Protocol/Internet Protocol ("TCP/IP"). SSL is used by HTTPS as access methods. Unfortunately, SSL

requires third party authentication, embedded certificates and exchange of certificates when the host name is changed. For these and other reasons, it is not suitable for management of embedded network appliances or certain other environments in which secure communication is desirable.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0005] **Figure 1** is a block diagram of the system of one embodiment of the invention.

[0006] **Figure 2a** is a schematic diagram of the frames sent from a Device Under Management in one embodiment of the invention.

[0007] **Figure 2b** is a schematic diagram of an exemplary concatenated string for one embodiment of the invention after decryption.

[0008] **Figure 3** is flow diagram of operation on the Device Under Management in one embodiment of the invention.

[0009] **Figure 4** is a flow diagram of operation on external node in one embodiment of the invention.

DETAILED DESCRIPTION

[0010] **Figure 1** is a block diagram of the system of one embodiment of the invention. A Device Under Management ("DUM") 100 is coupled to a distributed network such as internet 102. External node 104 is also coupled to internet 102. External node 104 may be used to manage DUM 100 over the internet 102. There may be an arbitrarily large number of DUM's coupled with the internet 102 up to DUM_N 108 with each device manageable by external node 104.

[0011] In one embodiment, external node 104 may be any internet access device that supports Java, Java Script, frames and forms. In one embodiment, external node 104 may be a personal computer (PC) executing a web browser such as Microsoft Explorer® or Netscape Navigator®. As previously noted, such browsers support Java, Java Script, frames and forms. DUM 100 may be any network element including an embedded network appliance, such as the InterJak™ 200 available from Filanet Corporation of Sunnyvale, California. DUM 100 may provide web server functions, or for example, a fire wall for clients 106.

[0012] When external node 104 first requests secure access to DUM 100, such as, for example, management access, device 100 serves a frame containing an embedded security applet to external node 104. In one embodiment, the security applet is a Java applet. In another embodiment, the security applet may be realized using embedded Java Script code. The security applet generates a login window on the external node 104. A user can then enter their required login data to gain access to the DUM 100. As used herein, "login data" may include a user I.D., a password, or

both. The security applet encrypts the login data and sends the encryption login data over the internet 102 to DUM 100. In one embodiment, the login data is encrypted with a random key generated in DUM 100 and sent to the external node as part of the Java Script code within the login page. DUM 100 decrypts the login data and determines if the login data is valid. If the login data is valid, it is used as a basis for a key for all subsequent encryption. As used herein to serve "as a basis for the key" is deemed to include, without limitation, direct usage as the key, usage of all or part of the login data as a seed for a pseudo random number generator to generate a pseudo random key, and indirect use, such as using the login data to encrypt a known data set and using the encrypted known data set as the key for subsequent encryption or using a hash of the login data as the key. When the login data is used directly, longer login data will yield stronger encryption.

[0013] Subsequent pages are provided to the external node 104 as subframes of the frame containing the security applet. Such subframes include the form having blank fields, a concatenated string of the field values with a digital signature all separated by appropriate delimiters and encrypted using the login data based key, and a script to decode and distribute the string. In one embodiment, the script is a Java Script. In one embodiment, 3DES (3 Data Encryption Standard) encoding is used. In one embodiment, the encoded string is transferred into the Java Script section of the web page together with a Java Script to decode the string and distribute it to the fields appropriately. If the user modifies the fields at the external node 104, the script provided as part of the page will concatenate at least the modified fields, digitally sign and encrypt the concatenated string. In one embodiment, only the

modified fields are concatenated. In an alternative embodiment, all field values are concatenated if any field is modified. In one embodiment, the Java Script hooks into the security applet which performs the heavy lifting of the encrypting and signing function.

[0014] **Figure 2a** is a schematic diagram of the frames sent from a Device Under Management in one embodiment of the invention. Parent frame 200, which has resident security applet (not shown), remains active on the external node for the entire secure session. As explained below, by retaining login data and using that data as a key basis for encryption and decryption, secure communication can be accomplished without third party authentication, embedded certifications, and change in certificate as the host name changes or other short comings of Secure Socket Layer ("SSL"). A subframe 202 includes blank form 212, which has blank fields 204, 206 and 208 as well as text or other static information 210. In one embodiment, the static information 210 is not encrypted and is readily discernable from the source code for the frame. In another embodiment, static text such as status information is also encrypted either as part of a single concatenated string or as separate strings. A submit button 214 may also be provided to permit the user to indicate that modifications to the fields should be sent back to the DUM.

[0015] **Figure 2b** is a schematic diagram of an exemplary concatenated string for one embodiment of the invention after decryption. A type and data length field delimits the data to be inserted in the plurality of blank fields. In this example, the content for field 204 ("204'") is a four byte integer value, 206' is a sixty-four byte

character value and 208′ is a single byte Boolean value. Other ways of delimiting the field contents are within the scope and contemplation of the invention.

[0016] **Figure 3** is flow diagram of operation on the Device Under Management in one embodiment of the invention. At functional block 302, the device receives a page access request that would be subject to secure access requirements. At functional block 304, the device forwards a frame with an embedded security applet to the requesting node. At functional block 306, the device receives and decrypts login data from the requesting node. The determination is made at decision block 308 whether the login data is valid. If the login data is not valid, access is denied and no page is sent.

[0017] If the login data is valid, the device builds a blank form at functional block 310. The field values for the form evident from the source code are "empty." Meanwhile, the field contents for which security is desired are concatenated into a string at functional block 312. At functional block 314, a digital signature is appended to the string. In one embodiment, the digital signature is acquired by forming a one-way hash of the string of concatenated field contents. The aggregate string, including the digital signature, is encrypted at functional block 316 using the login data as the key basis. A script, such as an appropriate Java Script, which may be tailor made for each page and the encrypted string may be appended to the blank form at functional block 318. The subframe including the blank form, the encrypted string, and the script is transmitted to their requesting node at functional block 320.

[0018] A determination is made at 322 whether a subsequent encrypted string is received from the requesting node at decision block 322. If such subsequent string is

received, the DUM decrypts the string and modifies its management values consistent with the values contained in the string at functional block 324. If no subsequent encrypted string is received, no further action is taken by the DUM, unless a further secure request is received, in which case a DUM continues executing from functional block 310 to provide the next requested form.

[0019] **Figure 4** is a flow diagram of operation on external node in one embodiment of the invention. At functional block 400, the external node requests a secure page from a network element, such as DUM 100 (referring to Figure 1). At functional block 402, the external node receives a frame with an embedded security applet. The security applet within the received frame generates a login window at functional block 404. Login data entered by the user is retained by the security applet and forwarded to the device at functional block 406. In one embodiment, the login data is encrypted by the security applet prior to being forwarded to the device. In another embodiment, the login data is hashed prior to encryption and the hash value is encrypted with the login data and forwarded to the device. At functional block 408, the external node receives a frame including a blank form (e.g., a form having blank fields), a script, and an encrypted string. In one embodiment, the script is a Java Script. At functional block 410, the script activates the security applet to decrypt the string using the previously retained login data as the key basis. At functional block 412, the script uses the applet to check the digital signature appended to the string. At functional block 414, the script distributes the string to the fields of the form. In one embodiment, the script parses the string based on delimiting fields denoting data length and data type.

[0020] At decision block 416, a determination is made whether the user has modified any fields of the form that should be submitted back to the device. In one embodiment, the determination is based on a submit event, such as actuation of a submit button within the subframe. If the user has modified the fields of the script within the subframe concatenates the field contents into a string at functional block 418. The script then causes the applet to sign and encrypt the string (using the login data based key) at functional block 420. At functional block 422, the encrypted string is returned to the device.

[0021] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.